De La Salle University
Computer Technology Department

STDISCM

Parallel Programming Project – Bulk Image Enhancer Project

Description

An image can be enhanced by adjusting brightness, contrast or sharpness. Image enhancing can make an image better for viewing. Enhancing a bulk number of images one by one with a program is cumbersome and slow. A parallel-processing based program can speed-up the process of enhancing bulk images.

Project Requirement

The following are the requirements for the project:

- Create a program that enhances bulk images in a specific amount of time
  - Input argument of the program are the following:
    - Folder location of images
    - Folder location of enhanced images
    - Enhancing time in minutes
    - Brightness enhancement factor
    - Sharpness enhancement factor
    - Contrast enhancement factor
    - Optional: Number of threads / process to use
  - Output of the program:
    - Enhanced images in the specified folder location
    - Text file that contains statistics of the processed files: number of images enhanced, output folder location,
- Set of reference images are to be used for processing
  - Images can be in jpg, gif or png format
- Program implementation:
  - Program can be implemented using any programming language
  - Program implementation should use parallel programming techniques
  - Program implementation can use libraries or APIs for image enhancement

Project Rubrics

The project is to be graded using the following criteria / rubric:

CRITERIA	EXEMPLARY 4	SATISFACTORY 3	DEVELOPING 2	BEGINNING 1
Technical Documentation 10 %	Document has presented the architecture of the system, pointed out the concepts, has given an excellent analysis of the performance of the system and provided a conclusion.	Document has presented the architecture of the system, pointed out the concepts, has given simple analysis of the performance of the system.	Document has presented the architecture of the system and pointed out the concepts.	No documentation
Parallel Techniques 40 %	Multiple processes or threads are used to distribute workload by using synchronization or parallel techniques	Multiple processes / threads are used but workload was not distributed nor use parallel techniques	Program essentially uses serial programming techniques	No program submitted
Performance 50%	Project is able to achieve task and result of task is consistent		Project is able to achieve required task but not done in parallel manner or result is not consistent	Project is not working

Documentation

Documentation requirements for the project is as follows:

- Document should have the outline:
  1. Introduction
     - Give a brief discussion of the project and its requirement
  2. Program Implementation
     - Discussion on how the program was implemented
       - Use of lock or semaphore objects
       - Sharing of data between processes
       - Parallel programming and optimization techniques used
  3. Result
     - Discussion of the results
     - Explanation or analysis why such results was achieved
  4. Conclusion
     Discuss briefly how parallel programming techniques was used
     Discuss how parallel programming techniques improved (or not improved) performance
  5. References
     - References used for concepts, programming techniques or libraries used

- Document is to follow the IEEE manuscript template for conference proceeding
  - Format for the manuscript is found at: IEEE - Manuscript Templates for Conference Proceedings

Submission Requirements

For submission:

- Document report
- Program Source Code
- Program output file (Multiple samples to show performance)
- Screenshots (If needed)

Deadline: 4th week November (Tentative)